



ROADMAP INGENIERO DE CALIDAD DE SOFTWARE DE PRIMER NIVEL

El siguiente documento contiene una serie de pasos a seguir para desarrollarte como Ingeniero de Calidad de Software. La información ha sido recabada de distintas fuentes y de nuestra experiencia en el área.

Sabemos que es posible profundizar en cada uno de los puntos detallados, pero hemos optado por dejar los principales lineamientos que corresponden a cada área de estudio.

Por último, no buscamos con esta Guía ser fuente indiscutible de programa a seguir, simplemente les ofrecemos un panorama más claro de lo que pueden alcanzar cada uno de ustedes.

INICIO – ALGUNAS HABILIDADES REQUERIDAS

- Curiosidad
- Pensamiento lateral y creativo
- Observación
- Comunicación oral y escrita
- Pensamiento crítico
- Adaptación al cambio
- Proactividad
- Trabajo en equipo

FUNDAMENTOS DEL TESTING

- Definición de Calidad
- Qué es el Testing y QA
- Objetivos, ¿para qué testeamos?
- Siete principios de las pruebas
- Oráculo de prueba
- Escenarios o casos preliminares de prueba.
- Caso de pruebas y técnicas: Clases de equivalencia, Valores límite, Tabla de decisión, Transición de estados, etc
- Pruebas basadas en la experiencia: Pruebas exploratorias, heurísticas de pruebas
- Priorización de pruebas
- Tipos de Defectos y Gestión de defectos

- Niveles de pruebas: Unitaria, Integración, de Sistema, UAT
- Tipos de prueba: Caja Negra, Caja Blanca, Pruebas funcionales y no funcionales, Pruebas asociadas al cambio (Regressión y Retesting), Testing negativo y positivo, Estático y Dinámico
- Documentación de Pruebas: Plan de pruebas, Casos de prueba, Mapas mentales, Cartas de prueba exploratoria, Resumen y avance de pruebas.
- Productos de trabajo: Requerimientos, Especificaciones, Casos de uso, Épicas, Historias de usuario, Diagramas de arquitectura, Diagrama de flujo de datos, Swagger
- Pruebas estáticas: Revisiones sobre productos de trabajo, Análisis estático sobre código, Tipos de revisiones.
- STLC: Planificación, Monitorización y Control, Análisis, Diseño, Implementación, Ejecución y Compleción de la prueba.
- Tipos de prueba no funcionales
- Ambientes y su gestión: Desarrollo, Testing, Staging, Producción

FUNDAMENTOS DEL DESARROLLO

- Modelos de desarrollo de software históricos: Cascada, Model V, Espiral, Iterativo e incremental
- Ventajas y desventajas de cada modelo.
- Niveles y actividades.
- Roles y responsabilidades
- Agile flavors: Scrum, Kanban, Lean
- Rol del QA en Agile

BUSCAR APOYO/CRECIMIENTO

- Blogs/sitios web
- Podcast
- Canales de YouTube
- Comunidades
- Seguir referentes en Twitter y LinkedIn

FUNDAMENTOS DE INTERNET Y APLICACIONES WEB

- Navegadores, motores de búsqueda
- Diferencia entre sitio web, página web, aplicación web
- Aplicaciones web y lenguajes: Html, css, javascript
- Web responsive vs reactive
- Seguridad, redes, arquitectura y protocolos
- AJAX
- SPA, PWA
- Cliente-servidor

- Frameworks: React, Angular, etc
- CMSs

FUNDAMENTOS DE CIENCIAS COMPUTACIONALES

- Hardware
- Datos: Binarios, Json, Xml, Yaml
- Lenguajes de programación: Alto y bajo nivel, Compilados e interpretados, Funcional y procedural, Orientado a objetos.
- Compiladores, depuradores.
- Tipos de sistema: Estáticos y dinámicos, De tipado fuerte y débil
- Estructura de datos
- Concurrency y Threading
- Sistemas operativos

PROGRAMACIÓN

- Command-line (CLI) vs Shell: Bash, PowerShell
- IDEs: VSCode, Sublime, PyCharm, IntelliJ
- Sintaxis del lenguaje de programación, operadores, flujo de control, estilos
- Patrones de diseño
- Programación funcional: Función de orden superior, Función pura, Objetos inmutables
- POO: Herencia, Abstracción, Polimorfismo, Acoplamiento, Cohesión Encapsulamiento
- Aprender algún lenguaje: Javascript, TypeScript, Java, C#, Python, etc.

ARQUITECTURA

- Arquitectura cloud: IaaS, PaaS, SaaS
- Servicios APIs, REST, GraphQL
- Microservicios
- Serverless
- Multitier
- Serialización: JSON, XML
- Comunicación síncrona y asíncrona
- Proxies
- Seguridad
- Logging y Monitoreo
- Proveedores cloud: AWS, GCP, Azure
- Base de datos: SQL vs NoSQL, Keys, SP, Views, Querys

FUNDAMENTOS DE TEST AUTOMATION

- Pirámide de testing
- ROI Automation
- Tipos de automation: Unitarios, API's, Web UI, Mobile, Desktop, Performance.
- Gestión de datos de prueba
- Frameworks
- Mocking, Stubbing: WireMock, Mountebank
- Low code vs no-code automation
- Record & Playback
- BDD/Gherkin: Cucumber, Speckflow
- Visual regression: Applitools, Percy

MODELO ÁGIL

- Scrum, Kanban, Lean
- Técnicas de estimación
- Gestión de proyectos ágiles: Jira, Rally, MS Project
- Ceremonias: Sprint planning, Daily, Sprint review, Sprint retrospective, Refinement
- HU y CA
- Scrum Team
- Rol y responsabilidad del QA: Ejemplos de approaches, Quality Gates, QA vs QC
- Objeciones al modelo ágil

UNIT TESTING AUTOMATION

- TDD
- Funcional y OO Unit
- Mocking y Code isolation
- Mutation Testing
- Frameworks según lenguaje de programación:
 - Java: JUnit, TestNG, NUnit, Mockito, EasyMock
 - JavaScript: Jest, Jasmine, Mocha
 - Python: PyTest

API AUTOMATION

- Frameworks: Postman, SoapUI, RestAssured, Karate, Katalon
- Contract Testing: Pact.io, Swagger

WEB UI AUTOMATION

- Limitaciones de UI automation
- Codeless, Record Playbak, Web Automation: Ranorex, Katalon, TestComplete
- Headless browsers
- Page Object Model, UI Patterns
- Frameworks para interfaz de usuario web

- No Selenium: Cypress, Puppeteer
- Basados en Selenium
- Diseño de frameworks
- Selectors
- Protocolo Web Driver
 - WebDriver.io
 - Protactor
 - Watir

CI/CD

- Estrategías de branching: Trunk, Feature, GitFlow, GitHubFlow
- Collaboration, Pull request
- Hosting Platforms: GitHub, BitBucket, AWS CodeCommit
- Configuration and Build
- Containers: Dockers, Kubernetes
- Integration with Automation
- Feature flags, Toggles
- Deployment and Releases
- Remote execution / Device Farms: SauceLabs, BrowserStack, AWS Device farm
- Orchestration: Jenkins, CircleCI, AWS CodePipeline, TeamCity

PERFORMANCE TESTING

- Taxonomía: Carga, Estrés, Resiliencia, Resistencia, Pico
- Performance en Pipeline
- Performance en Agil
- Herramientas: JMeter, Gatling, K6

MOBILE TESTING

- Apps Nativas, Híbridadas, Mobile web
- iOS
- Android
- Frameworks multiplataformas: Xamarin, React Native, Titanium
- Device farms
- Emulador vs Simulador
- Herramientas Mobile automation: Appium, XCode, Calabash, Espresso, Robotium
- Tipos de pruebas: Configuración, Performance, Compatibilidad del dispositivo con los SO, Conexión, Seguridad, etc.

ACCESSIBILITY TESTING

- Estándares 508
- WCAG2
- Herramientas de escaneo: Chrome DevTools Audit, Wave, SortSite, Axe

SECURITY TESTING

- Authentication vs Authorization
- Attack Vectors
- Surface Area
- Penetration Testing
- OWASP Top 10
- Vulnerability Scanners